Proceedings of the 7th IASTED International Conference
SOFTWARE ENGINEERING AND APPLICATIONS
November 3-5, 2003, Marina Del Rey, CA, USA

Assessing Usability of a Risk-based Requirements and Design Tool

James D. Kiper
Computer Science &
Systems Analysis
Miami University
Oxford OH 45056 USA
kiperjd@muohio.edu

Brent Auernheimer
Computer Science
California State University
Fresno CA 93740 USA
brent@csufresno.edu

Martin S. Feather
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Dr
Pasadena CA 91109-8099
Martin.S.Feather@jpl.nasa.gov

Abstract

This paper describes some work on assessing the usability of a tool for requirements analysis and design that is being designed and used at NASA and JPL. We describe the tool and its associated, risk-driven method. Then we present some usability questions and methods for answering these.

Key Words

Usability testing, cognitive walkthrough, requirements, design, risk

1. Introduction

The system requirements analysis and design process is an iterative one that requires much user involvement and interaction. In providing user support for this process, an effective user interface is vital. Data collection clearly depends on the judgments of experienced engineers and domain experts. Automation can help in analysis of data, but the results require human scrutiny.

In this paper, we briefly describe a method of risk-based requirements analysis and system design, and a tool that supports this process. Then we examine usability questions about this tool, and discuss ways of answering these questions.

2. Risk-based requirements and design

2.1 The problem

The process of collecting requirements and producing an appropriate design of a complex system is difficult at best. Spacecraft development exemplifies the challenges inherent in these activities. Spacecraft are complex devices, the development of which requires addressing cross-disciplinary concerns, and making many trade-off decisions. Budget and schedule pressures constrain their development. Wise choices among available design alternatives and quality assurance activities have to be made.

2.2 The tool and method

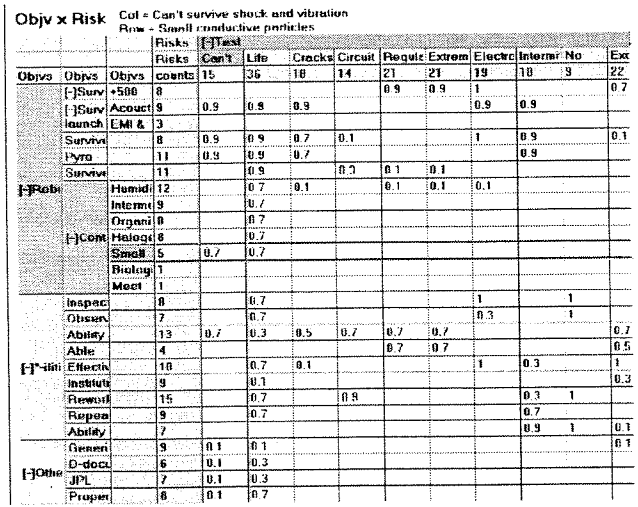
The approach to system requirement analysis and design described here is a risk-based approach called *Defect Detection and Prevention* (DDP) supported by a custom-built software tool [1,2]. It has been used at JPL to assist several projects aiming to infuse new technology into spacecraft systems, and is in ongoing use for risk management on an entire mission.

The DDP model includes three primary concepts: *requirements*, *risks*, and *mitigations*. The term *requirements* has the usual meaning – a description of those functions that the completed system must be able to perform, the non-functional constraints on the ways that it achieves these functions, and constraints on the development process of that system. *Risk* is anything that keeps the system from achieving its requirements. This might include development risks as well as operational risks. A *mitigation* is any activity or tool that lowers the probability of at least one risk. For a software system, mitigations could include inspections, testing, use of formal methods of specification, etc. For hardware aspects of spacecraft design, mitigations could be inspection of solder joints, use of packaging materials to protect components in space, etc.

The goal of the DDP process is to collect information about requirements, associated risks, and possible mitigations from the engineers and domain experts who know the most about them; and to represent this data in a form that managers and engineers can use to make judgments that best reduce the risk in implementing this system.

To help reason about this model, the engineers and domain experts are asked to make judgments about the importance of each requirement. This is recorded as the requirement's *weight*. Similarly, these engineers and experts are asked to assign an a priori *probability* of each risk's occurrence, and an estimate of the *cost* of each mitigation.

Requirements, risks and mitigations are linked through two matrices. The first of these matrices correlates risks and requirements. (See figure 1.) Engineers and domain experts are asked to estimate the impact that each risk, should it occur, would have on each requirement. Impacts are expressed as numbers in the range 0 to 1. An estimate of 1 means that this risk is so serious that, if it occurs, that requirement would be completely lost. A value of 0 implies that this risk will not effect this requirement at all. A value of ρ (0 < ρ < 1) means that the experts expect that risk to cause the proportion ρ of that requirement to be lost. It is also possible to provide a non-numeric value, which will be ignored in the calculations performed by the DDP tool, but which has utility as a placeholder (e.g., a "TBD" value which someone will be responsible for looking up).



Objv x Risk Col = Can't survive shock and vibration
Row = Small conductive particles

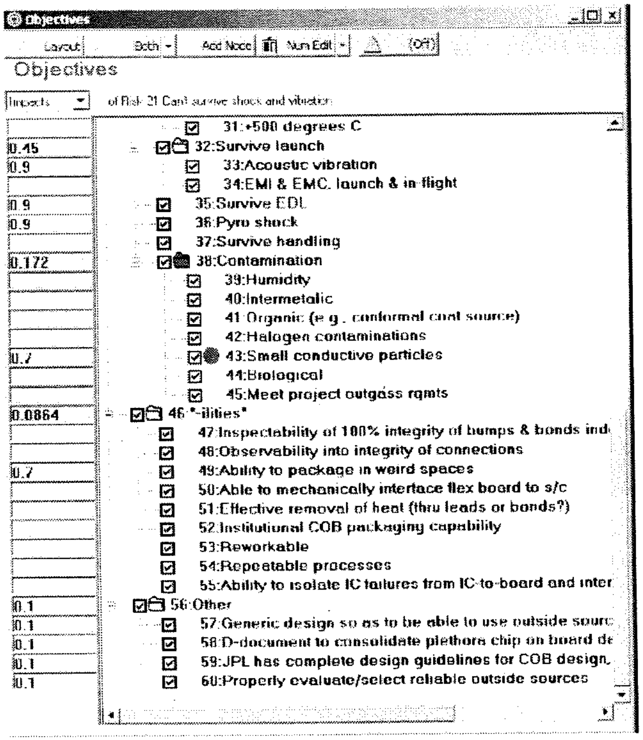
			Risks	[-]Test									
			Risks	Can't	Life	Cracks	Circuit	Requir	Extrem	Electrc	Intermr	No	Exc
Objvs	Objvs	Objvs	counts	15	36	18	14	21	21	19	18	9	22
	[-]Surv	+500	8					0.9	0.9	1			0.7
	[-]Surv	Acouct	9	0.9	0.9	0.9				0.9	0.9		
	launch	EMI &	3										
	Survive		8	0.9	0.9	0.7	0.1			1	0.9		0.1
	Pyro		11	0.9	0.9	0.7					0.9		
	Survive		11		0.9		0.3	0.1	0.1				
[-]Robi		Humidi	12		0.7	0.1		0.1	0.1	0.1			
		Interme	9		0.7								
		Organi	8		0.7								
	[-]Cont	Halog	8		0.7								
		Small	5	0.7	0.7								
		Biolog	1										
		Meet	1										
	Inspec		8		0.7					1		1	
	Obser		7		0.7					0.3		1	
	Ability		13	0.7	0.3	0.5	0.7	0.7	0.7				0.7
	Able		4					0.7	0.7				0.5
[-]*-ilit	Effecti		10		0.7	0.1				1	0.3		1
	Institut		9		0.1								0.3
	Rework		15		0.7		0.9				0.3	1	
	Repea		9		0.7						0.7		
	Ability		7								0.9	1	0.1
	Generi		9	0.1	0.1								0.1
	D-docu		6	0.1	0.3								
[-]Othe	JPL		7	0.1	0.3								
	Proper		8	0.1	0.7								

Figure 1: Objective by risk

Figure 2 shows a list view of objectives that can be used to input impacts for one specific risk. In this view, the names of objectives are more readable.

The second matrix correlates risk and mitigations. That is, for each mitigation-risk pair, domain experts are asked to estimate the ability of that mitigation to alleviate, detect or prevent that risk. Again, these are numbers in the range 0 to 1, with 1 meaning that this mitigation completely eliminates this risk; 0 means that it has no effect on this risk. (For mitigations that *detect* risks, the intent is that these be applied before the spacecraft is launched, in time for any detected problems to be repaired).

Up to this point in the DDP process, the primary task of the DDP tool is to *record* these decisions. After this data is collected, the focus of the process switches to one of decision making, guided by the recorded information. The primary purpose of DDP has been to help guide the selection of which of the mitigating actions to take to overcome the risk and therefore to achieve the requirements. Since there are typically many more possible mitigations than can be simultaneously afforded, the aim of this step is to emerge with a cost-effective selection from among them. Another outcome can be the decision to modify and/or abandon some requirements if it becomes clear that it is not possible to satisfactorily achieve them all with the resources available. This is called *descoping* [3].



Objectives

Impacts of Risk 21 Can't survive shock and vibration

	☑ 31:+500 degrees C
0.45	☑ 32:Survive launch
0.9	☑ 33:Acoustic vibration
	☑ 34:EMI & EMC. launch & in-flight
0.9	☑ 35:Survive EDL
0.9	☑ 36:Pyro shock
	☑ 37:Survive handling
0.172	☑ 38:Contamination
	☑ 39:Humidity
	☑ 40:Intermetalic
	☑ 41:Organic (e.g., conformal coat source)
	☑ 42:Halogen contaminations
0.7	☑ ● 43:Small conductive particles
	☑ 44:Biological
	☑ 45:Meet project outgáss rqmts
0.0864	☑ 46:"-ilities"
	☑ 47:Inspectability of 100% integrity of bumps & bonds ind
	☑ 48:Observability into integrity of connections
0.7	☑ 49:Ability to package in weird spaces
	☑ 50:Able to mechanically interface flex board to s/c
	☑ 51:Effective removal of heat (thru leads or bonds?)
	☑ 52:Institutional COB packaging capability
	☑ 53:Reworkable
	☑ 54:Repeatable processes
	☑ 55:Ability to isolate IC failures from IC-to-board and inter
0.1	☑ 56:Other
0.1	☑ 57:Generic design so as to be able to use outside sourc
0.1	☑ 58:D-document to consolidate plethora chip on board de
0.1	☑ 59:JPL has complete design guidelines for COB design,
0.1	☑ 60:Properly evaluate/select reliable outside sources

Figure 2: Objectives – list view.

In addition to the matrix and list views used primarily for input of data, DDP provides several views appropriate for study of various measures calculated from this data. One of these is a familiar bar chart view. For example, when showing the aggregate impact that each risk has on the requirements (taking in to account both the weight of the requirements and the strengths of the impacts), this is useful to indicate the "tall pole" risks worthy of further attention. (See figure 3 below.)

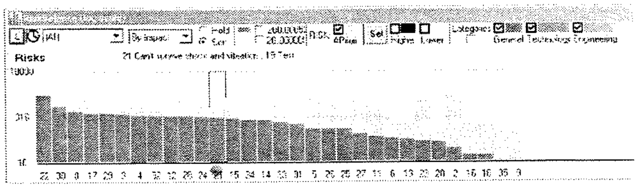
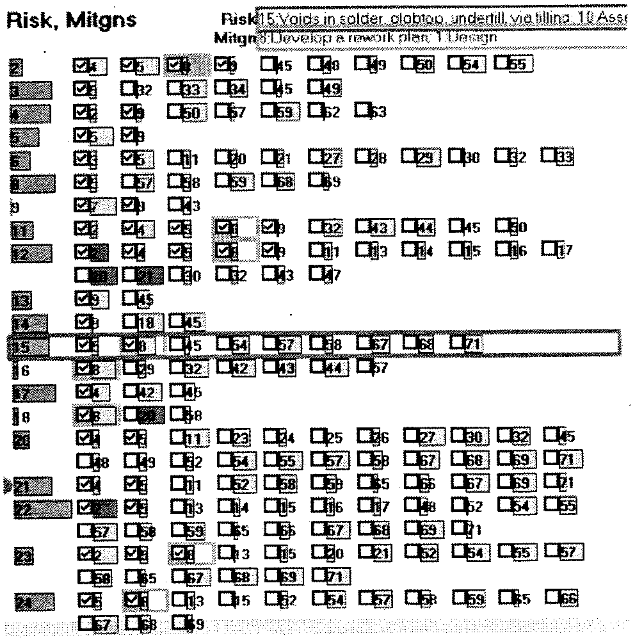


Figure 3: Risk bar chart

Figure 4 reveals a portion of another view in which each row corresponds to a risk and the mitigations that reduce it. Clicking on one of these mitigations causes all instances of that mitigation to flash. This enables a manager to see the overall effect of particular mitigations.



Risk, Mitgns

Risk 15:Voids in solder, globtop, underfill, via filling; 10:Ass

Mitgn 8:Develop a rework plan; 1:Design

2 ☑4 ☑5 ☑8 ☑9 ☐45 ☐48 ☐49 ☐50 ☐54 ☐55
3 ☑3 ☐32 ☐33 ☐34 ☐45 ☐49
4 ☑2 ☑9 ☐50 ☐57 ☐59 ☐62 ☐63
5 ☑5 ☑9
6 ☑3 ☑5 ☐11 ☐20 ☐21 ☐27 ☐28 ☐29 ☐30 ☐32 ☐33
8 ☑3 ☐57 ☐58 ☐59 ☐68 ☐69
9 ☑7 ☑9 ☐43
11 ☑2 ☑4 ☑5 ☑8 ☑9 ☐32 ☐43 ☐44 ☐45 ☐50
12 ☑2 ☑4 ☑5 ☑8 ☑9 ☐11 ☐13 ☐14 ☐15 ☐16 ☐17
☐20 ☐21 ☐30 ☐32 ☐43 ☐47
13 ☑9 ☐45
14 ☑8 ☐18 ☐45
15 ☑5 ☑8 ☐45 ☐54 ☐57 ☐58 ☐67 ☐68 ☐71
16 ☑8 ☐29 ☐32 ☐42 ☐43 ☐44 ☐57
17 ☑4 ☐42 ☐45
18 ☑8 ☐20 ☐58
20 ☑4 ☑5 ☐11 ☐23 ☐24 ☐25 ☐26 ☐27 ☐30 ☐32 ☐45
☐48 ☐49 ☐52 ☐54 ☐55 ☐57 ☐58 ☐67 ☐68 ☐69 ☐71
21 ☑4 ☑5 ☐11 ☐52 ☐58 ☐59 ☐65 ☐66 ☐67 ☐69 ☐71
22 ☑2 ☑5 ☐13 ☐14 ☐15 ☐16 ☐17 ☐48 ☐52 ☐54 ☐55
☐57 ☐58 ☐59 ☐65 ☐66 ☐67 ☐68 ☐69 ☐71
23 ☑2 ☑5 ☑8 ☐13 ☐15 ☐20 ☐21 ☐52 ☐54 ☐55 ☐57
☐58 ☐65 ☐67 ☐68 ☐69 ☐71
24 ☑5 ☑8 ☐13 ☐15 ☐52 ☐54 ☐57 ☐58 ☐59 ☐65 ☐66
☐67 ☐68 ☐69

Figure 4: Risks and their mitigations

Figure 5 presents a portion of a tree graph view of requirements. This uses nesting of rectangles to show hierarchy. Then color (shading degree) is used to show the degree to which each requirement is under risk; rectangle size captures the user assigned importance weight of that requirement.

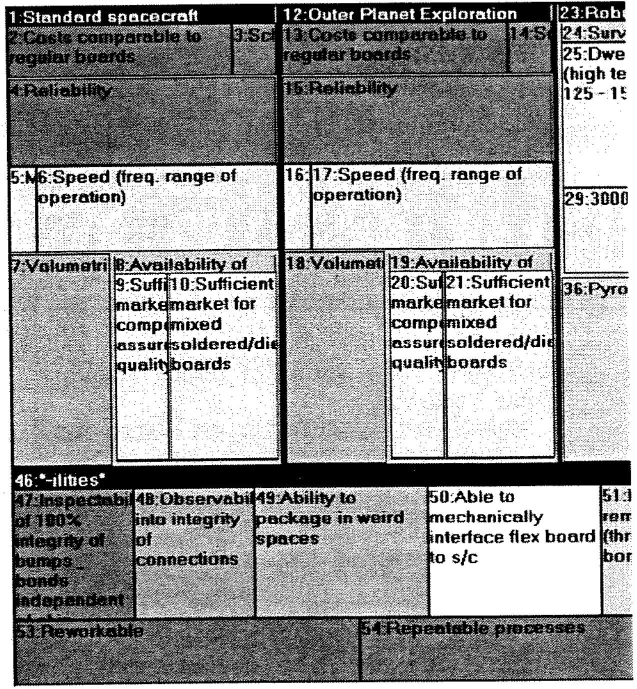

1:Standard spacecraft
2:Costs comparable to regular boards
3:Sci
4:Reliability
5:M
6:Speed (freq. range of operation)
7:Volumetri
8:Availability of
9:Suffi marke comp assur quality
10:Sufficient market for mixed soldered/die boards
12:Outer Planet Exploration
13:Costs comparable to regular boards
14:S
15:Reliability
16:
17:Speed (freq. range of operation)
18:Volumet
19:Availability of
20:Suf marke comp assur quality
21:Sufficient market for mixed soldered/die boards
23:Rob
24:Surv
25:Dwe (high te 125 - 15
29:3000
36:Pyro
46:*-ilities*
47:Inspectabi of 100% integrity of bumps bonds independent
48:Observabi into integrity of connections
49:Ability to package in weird spaces
50:Able to mechanically interface flex board to s/c
51: rem (thr bor
53:Reworkable
54:Repeatable processes

Figure 5: Tree graph of requirements.

Another useful view of risk is the impact-likelihood "fever" graph as shown in figure 6. Each point of this graph represents a risk. In the tool, when the user clicks on a point, the name of that risk is displayed. Those risks whose corresponding points graph to the upper right corner are generally ones that present the highest priorities for managers to address.

It is typical for DDP studies to involve a non-trivial amount of data. There are commonly dozens (or even hundreds) each of requirements, risks and mitigations, and there may well be thousands of non-zero impact and effect values connecting these.

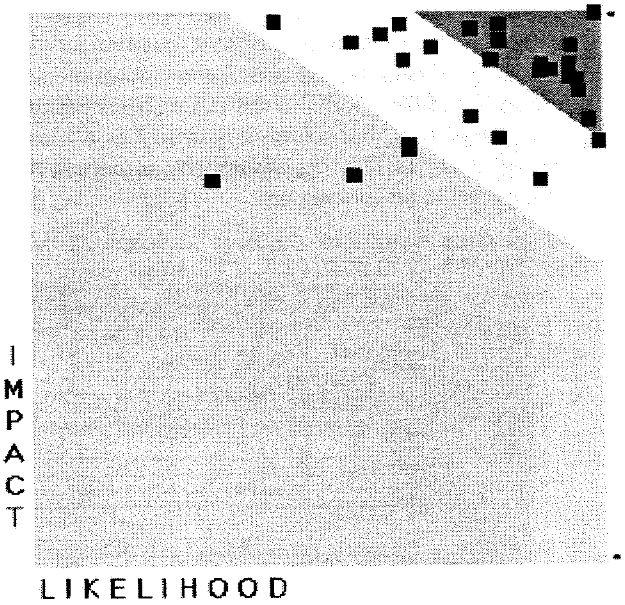


Figure 6: Impact versus likelihood.

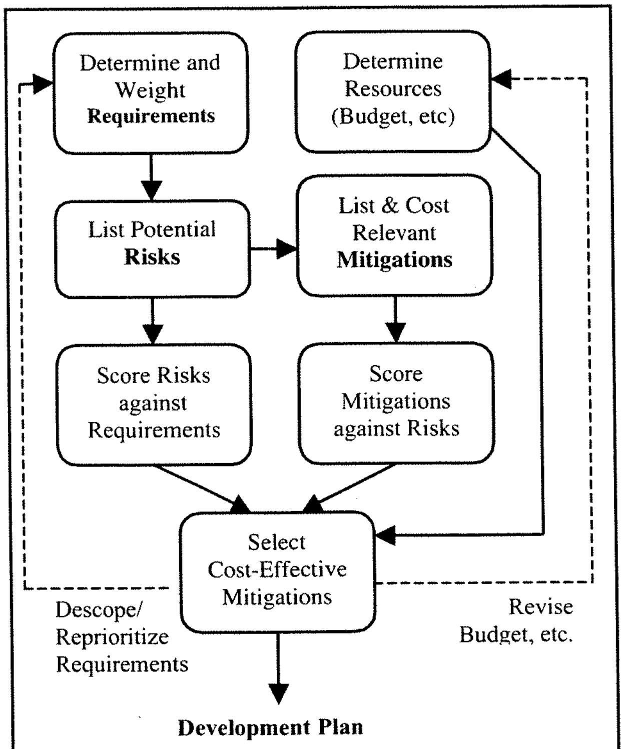
The volume and interconnectedness of this data reflects the complexity of the challenges inherent in the design of complex systems. DDP aims to help in this design process, and so must manage this quantity of information in such a way as to allow the engineers and domain experts both to provide the information, and make decisions on the basis of that aggregated information.

3. Tool Use

3.1 A typical scenario

The DDP process is presented in Figure 7. Here we give a scenario of usage of the tool. A typical DDP session convenes with a group of five to fifteen engineers and domain experts meeting in a room in which the DDP data entry screens are projected. A moderator who is trained in the DDP process leads the meeting. A trained "driver" enters data into DDP. The DDP database is often pre-loaded with a first estimate of system requirements if these are available.

The moderator leads the group in eliciting system requirements. The tool allows these to be structured hierarchically. (One of the tasks of the moderator and driver is to look for commonalities and structure in the requirements.) As these requirements are collected, the experts are asked to give each a weight. This is an integer that specifies the importance of that requirement. (The range of values to be used is determined by the group. For example, they may choose to weight requirements from 1 to 100. Larger numbers always mean greater importance.)



Determine and Weight **Requirements**
Determine Resources (Budget, etc)
List Potential **Risks**
List & Cost Relevant **Mitigations**
Score Risks against Requirements
Score Mitigations against Risks
Select Cost-Effective Mitigations
Descope/ Reprioritize Requirements
Revise Budget, etc.
Development Plan

Figure 7: Standard DDP Process

When the requirements and their weights have been collected, the moderator directs the group to the process of listing risks associated with these requirements. As defined previously, a risk is anything that stands in the way of satisfying one or more requirements. As these are generated and recorded in the DDP tool, the session participants are asked to assess the a priori likelihood of occurrence of each risk. This is an iterative process in that a discussion of some risks may lead to additional requirements, or modifications to existing requirements.

When the risk enumeration seems to be complete, the DDP driver switches to the requirement × risk matrix. In this view, for every requirement f_i and risk r_j participants are asked to assess the proportion of the requirement f_i that risk r_j will lose should that risk occur. As described previously, this is a number in the range 0 to 1. The driver is, of course, recording these assessments in the tool.

At this point in the process, the moderator leads the group to a discussion of the mitigations that are possible to detect, alleviate, or avoid the risk. An estimate of the cost of each mitigation is recorded in DDP. When this list is relatively complete, the driver displays the risk × mitigation matrix. For each risk r_i and mitigation m_j the experts are asked to estimate the effectiveness that the mitigation m_j has in reducing or alleviating risk r_i. This is a number in the range 0 to 1 as described previously.

The part of the process described above typically takes three half-day sessions for studies of individual technologies. These kind of studies comprise the majority of the applications of DDP to date. DDP is also in use as a risk management tool for an entire mission, which extends over a much longer period of time.

After the data and its relationships are captured, the engineers, domain experts and managers use the DDP tool to scrutinize the totality of that data to help them in their decision making. Heuristic search methods (simulated annealing and genetic algorithms) are implemented within DDP to locate near-optimal selections of mitigations (e.g., those that maximize attainment of requirements while costing no more than a user-defined cost ceiling). While such methods help, the final decisions are made by the experts, informed by DDP via its calculations and visualizations.

Other views of the data allow the manager to see which requirements are being satisfied and which are still under large risk. This view may lead the manager to ask the project team to reduce the capabilities of the system. That is, to discard requirements or reduce their weights (relative importance).

The model of DDP involving requirements, risks, mitigations, and their relations may appear too weak for useful reasoning. However, in repeated sessions with DDP, it has been seen that the model is rich enough to structure and simplify debates between NASA experts. For example, DDP has been applied to over a dozen applications to study advanced technologies such as:

- a computer memory device
- gyroscope design
- software code generation
- a low temperature experiment's apparatus
- an imaging device
- circuit board like fabrication
- micro electromechanical devices
- a sun sensor; \item a motor controller
- photonics
- interferometry

In those studies, DDP sessions have found cost savings exceeding $1 million in at least two of these studies, and lesser amounts (exceeding $100,000) in the other studies. The DDP meetings have also generated numerous design improvements such as saving power or mass and shifting of risks from uncertain architectures to better understood designs. Further, at these meetings, some non-obvious, but significant risks have been identified and mitigated. Lastly, DDP can be used to document resolutions to those debates. Hence, DDP is in use at JPL.

3.2 Usability questions

As indicated in the previous section, the DDP process and tool is being used at JPL. However, it is clearly a developing technology. The primary users, i.e., drivers and moderators, are intimately involved in the design and development of this tool. The user interface works well for them. However, there are many open questions about how the effectiveness of this interface for others as usage expands across NASA and even outside. In this section we discuss some of the important usability questions to be answered.

3.2.1 Is DDP easy to learn for a "driver". To this point, the primary driver has been the primary tool designer and implementer. Use of the tool's interface requires mastery of several views – entry screens for requirements, risks, mitigations; matrices for requirements × risks and risks × mitigations; bar graphs to display risk strength; and others.

3.2.2 Is the information density on the screen appropriate for users? The DDP tool allows for multiple windows, has several menus along the top, includes many tool bar buttons. In its typical usage, it is common to have important information displayed in all areas of the screen. It is, of course, possible to hide some windows. What is the optimal amount of information to display?

3.2.3 Is the matrix form of the data useful for users? Two of the most often windows used in a DDP session are those that display the two matrices – requirements × risks and risk × mitigations. On a typical computer screen only a portion of these matrices can be seen at one time. It is necessary to scroll back and forth and up and down. When requirements and risks are structured in hierarchies (which is generally a good idea), it is difficult to display the complete names without using much screen real estate.

3.2.4 Is the default color scheme used throughout DDP helpful to users? The color scheme that displays requirements in blue, risks in red, and mitigations in green is use consistently throughout the tool. Is this effective, or do the colors overload the user?

3.2.5 Is it better to tile a single screen with multiple windows, or to have a "tabbed" interface with each tab being a different window? As described above, the information density of the screen can be quite high. Would it be more effective to use a tabbed interface that allows more screen space for each window, even though only one window is seen at a time?

3.2.6 Are the tool bar button icons meaningful to users? Are there too many buttons? The tool bar button icons have been carefully designed to convey a hint about their purposes to the user. Do other users understand this meaning?

3.2.7 Should there be a separate interface for the "driver" from that seen by meeting participants? In the current use process, the session participants and the tool driver view the same interface. Would it be more effective for the driver to see an interface that facilitates data input, with the session participants seeing a less cluttered view.

4. Method

Why not just ask users what they think of the interface? The answer is that self-reported data is notoriously unreliable. Nielsen gives some "basic rules of usability" [4]:

- ➢ Watch what people actually do.
- ➢ Do not believe what people *say* they do.
- ➢ Definitely don't believe what people predict they *may* do in the future.

Taking Nielsen's advice, we will apply several techniques to evaluate DDP usability. The two primary approaches will be *usability testing* and *cognitive walkthroughs*.

Usability testing involves having typical users performing prescribed tasks with the interface being evaluated. Subjective (Likert scale satisfaction, for example) and objective (time to complete tasks, number of errors) data are collected and analyzed. Nielsen characterizes usability testing as follows [5, p. 165]:

> User testing with real users is the most fundamental usability method and is in some sense irreplaceable, since it provides direct information about how people use computers and what their exact problems are with the concrete interface being tested.

Although testing with real users is expensive, substantial usability problems can be found using just a handful of users [6].

Second, we will apply cognitive walkthrough techniques to evaluate the interface. As described in [7] this technique is an

> information-processing model of human cognition that describes human-computer interaction in terms of four steps:
> 1. The user sets a goal to be accomplished with the system (for example, "check spelling of this document").
> 2. The user searches the interface for currently available actions (menu items, buttons, command-line inputs, etc.).
> 3. The user selects the action that seems likely to make progress toward the goal.
> 4. The user performs the selected action and evaluates the system's feedback for evidence that progress is being made toward the current goal.

One difference between these two techniques is that cognitive walkthroughs do not require user participation. Expert evaluators examine typical usage scenarios, explore possible user responses, and evaluate alternatives. This allows application of cognitive walkthroughs to be applied to incomplete interfaces, or in situation when users are not available for usability testing.

These two techniques are suitable to answer the questions described in section 3.2. For example, usability testing techniques could be applied to question 3.2.6 about meaningful icons and buttons. Typical interface tasks will be devised and user performance recorded. Of particular interest in this case will be number of user errors or misinterpretations of icon meanings.

The cognitive walkthrough technique can be applied to question 3.2.1 about easy of learning. Rieman et al. note [7] that

> The cognitive walkthrough is a technique for evaluating the design of a user interface, with special attention to how well the interface supports "exploratory learning," i.e., first-time use without formal training.

We anticipate that a significant amount of usability testing and cognitive walkthrough results will be available for reporting at the conference.

5. Summary

We've described a tool for evaluating risks - and their possible mitigations – during the design of complex systems. Because of the importance of these evaluations the tool's user interface should minimize user error and maximize user productivity. We outline several usability questions and discuss two particular techniques for answering these questions.

Acknowledgements

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology and at Miami University, Oxford, Ohio, both under a contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government, the Jet Propulsion Laboratory, California Institute of Technology, or Miami University.

Dr. Steve Cornford leads the DDP project at JPL. DDP technology has been developed with NASA's Software IV&V, Code R and Code Q support, managed in part through JPL's Assurance Technology Program Office.

References

1. Cornford, S. L., M. S. Feather, et al. (2001). DDP – A tool for life-cycle risk management. IEEE Aerospace Conference, Big Sky, Montana.
2. Feather, M.S., Cornford, S.L. Dunphy, J. & Hicks, K.A. (2002). A Quantitative Risk Model for Early Lifecycle Decision Making; Proceedings of the Conference on Integrated Design and Process Technology, Pasadena, California, June 2002. Society for Design and Process Science
3. Feather, M.S., S.L. Cornford & K.A. Hicks (2002) Descoping; Proceedings of the 27th IEEE/NASA Software Engineering Workshop, Greenbelt, Maryland, Dec 2002. IEEE Computer Society.
4. J. Nielsen. First Rule of Usability? Don't Listen to Users. http://www.useit.com/alertbox/20010805.html, 2001.
5. J. Nielsen. *Usability Engineering*. San Francisco: Morgan Kaufmann, 1993.
6. J. Nielsen. Why You Only Need to Test With 5 Users. http://www.useit.com/alertbox/20000319.html, 2000.
7. Rieman, J., Franzke, M. & Redmiles, D. Usability Evaluation with the Cognitive Walkthrough. http://www.acm.org/sigchi/chi95/Electronic/documnts/tutors/jr_bdy.htm